

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

BL2D-V2, un générateur de maillages dans le plan

DOCUMENT PROVISOIRE

Patrick Laug — Houman Borouchaki

N° ????

Octobre 2002

THÈME 4

Rapport technique

ISSN 0249-0803



BL2D-V2, un générateur de maillages dans le plan

DOCUMENT PROVISOIRE

Patrick Laug* , Houman Borouchaki†

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Gamma

Rapport technique n° ???? — Octobre 2002 — 54 pages

Résumé : Le logiciel BL2D-V2 génère des maillages dans le plan, isotropes ou anisotropes. Il peut être intégré dans un processus adaptatif. Ce rapport contient un manuel d'utilisation, un manuel de programmation, et quelques exemples d'applications de ce mailleur.

Mots-clés : maillage adaptatif, maillage anisotrope, métrique riemannienne, structure de données.

* E-mail : Patrick.Laug@inria.fr
† E-mail : Houman.Borouchaki@utt.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

BL2D-V2, a plane mesh generator

PROVISIONAL DOCUMENT

Abstract: The BL2D-V2 software package generates meshes in the plane, isotropic or anisotropic. It can be integrated within an adaptive process. This report includes a user's manual, a programmer's manual, and some application examples of this mesher.

Key-words: adaptive mesh, anisotropic mesh, Riemannian metric, data structure.

Table des matières

– Partie 3 : Exemples d'applications

Introduction

Présentation générale du logiciel

Le logiciel BL2D-V2 crée des maillages isotropes ou anisotropes dans le plan. Il peut être intégré dans un processus adaptatif. Son architecture logicielle est modulaire, et deux composants jouent un rôle central, le premier qui discrétise les courbes et le second qui maille les domaines du plan :

- **Discrétisation des courbes.** On suppose que la frontière Γ d'un domaine Ω de $\mathbb{R}^2$ est donnée sous une forme échantillonnée, constituée par un ensemble de segments droits. Cet échantillonnage est lissé par une spline cubique, qui est approchée avec une grande précision par un segment polygonal, afin de définir un support géométrique de la frontière Γ. Enfin, ce support géométrique est discrétisé en respectant les tailles spécifiées [1].
- **Maillage des domaines.** À partir de la frontière Γ discrétisée précédemment, le mailleur insère des points sur des arêtes internes au domaine Ω (méthode algébrique) ou par couches successives (méthode frontale), puis connecte ces points entre eux par une méthode de Delaunay généralisée [2, 3]. Il génère des triangles droits (P^1) ou courbes de degré 2 (P^2), ou encore des quadrilatères Q^1 ou Q^2 par appariement.

La construction directe d'éléments de degré 2 est rendue possible via le contrôle du maillage des frontières du domaine, de façon à assurer la compatibilité désirée. Les nœuds milieux frontières sont placés en fonction de l'abscisse curviligne. Les nœuds milieux internes sont placés de manière à optimiser la qualité en forme des éléments.

En outre, le logiciel BL2D-V2 traite le cas des domaines à frontières mobiles, dont la forme varie fortement au cours du temps.

Description fonctionnelle du logiciel

En entrée, un fichier (format *g*) contient les données géométriques et physiques de plusieurs sous-domaines du plan. Il peut être créé directement par un préprocesseur intégré (`blg`), ou être extrait de maillages donnés dans un autre format (*nopo, mesh, amdba, ms,* etc.). Un deuxième fichier d'entrée (format *h*) permet de spécifier la taille et la forme des éléments à générer.

En sortie, un nouveau fichier (format *ms*) contient le maillage demandé. Ce fichier peut également être converti en de nombreux autres formats.

Remarques

Le logiciel BL2D-V2 est parfois référencé ailleurs sous le nom de BLMESH.

Il est issu d'une ancienne version V1 datant de 1995 [4, 5]. Il est cependant très différent de sa version initiale, les principales nouveautés inclues dans la version V2 étant les suivantes : méthode frontale (placement optimal des points, meilleure qualité en forme des éléments), triangles de degré 2, quadrilatères de degré 1 ou 2, frontières mobiles, et robustesse accrue (en particulier, détection des auto-intersections de la frontière), etc.

À quelques détails près, les formats de fichiers de la version V2 sont identiques à ceux de la version V1.

Plan du rapport

Ce rapport est divisé en trois parties :

- un manuel d'utilisation,
- un manuel de programmation, plus spécialement destiné au développeur cherchant à interfacer, modifier ou étendre le logiciel BL2D-V2, et
- quelques exemples d'applications.

Partie 1 : manuel d'utilisation

Dans cette première partie, une initiation à l'utilisation du logiciel est proposée à partir de plusieurs exemples (section 1). Ensuite sont données par ordre alphabétique une description des différents programmes du logiciel (section 2) et celle des formats de fichiers utilisés (section 3).

1 Initiation au logiciel à partir d'exemples

L'objectif de cette section est d'apprendre rapidement à utiliser les différentes fonctionnalités du logiciel. À cette fin, un premier exemple simple est donné, suivi d'autres exemples avec des maillages adaptatifs et des maillages mobiles.

1.1 Exemple simple

Dans cet exemple simple, nous voulons mailler un triangle équilatéral de sommets A (-1, 0), B (1, 0) et C (0, $\sqrt{3}$). Il suffit pour cela de créer un fichier `x.0.0.data`[1] et de lancer le shell-script `bl2d.sh` (figure 1).

Fichier `x.0.0.data`:

```
100   100   100
A -1 0
B  1 0
C  0 1.732
;
;
AB NULL A B NULL ;
BC NULL B C NULL ;
CA NULL C A NULL ;
;
;
;
AB +1 0
;
isotrope
;
```

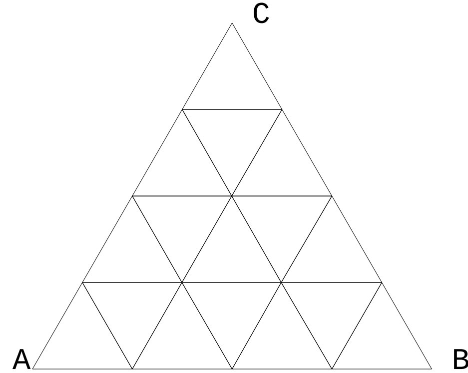


FIG. 1 – *Maillage uniforme.*

Donnons à présent quelques explications sur l'exemple précédent. Dans le fichier `x.0.0.data` (détaillé en section 3.2), les trois premiers entiers représentent un nombre nombre maximum

1. Un exemple de fichier `x.0.0.data` est fourni avec le logiciel BL2D-V2.

de points, de courbes et de sous-domaines définis dans ce fichier (ici 3 points, 3 courbes et 1 sous-domaine, d'où un majorant très large égal à 100 dans chaque cas). Ensuite, les sommets A, B, C et les côtés AB, BC et CA sont spécifiés (le mot-clé `NULL` signifie que les tangentes aux extrémités ne sont pas précisées). La ligne `AB +1 0` signifie que le domaine à mailler est à gauche du segment AB et que son numéro de référence est 0. Les éléments doivent être isotropes, c'est-à-dire le plus équilatéraux possible. Comme aucune taille n'est spécifiée, les segments droits qui constituent la frontière sont discrétisés avec une taille maximale (par défaut $1/5^e$ de la diagonale du rectangle englobant), et le maillage correspondant à cette taille est généré.

Pour obtenir un maillage uniforme plus fin, il est possible de spécifier une taille d'éléments aux points A, B et C (voir figure 2 avec une taille de 0.1).

Fichier `x.0.0.data`:

```
100   100   100
A -1 0
B  1 0
C  0 1.732
;
;
AB NULL A B NULL ;
BC NULL B C NULL ;
CA NULL C A NULL ;
;
;
;
AB +1 0
;
isotrope
A 0.1   B 0.1   C 0.1 ;
```

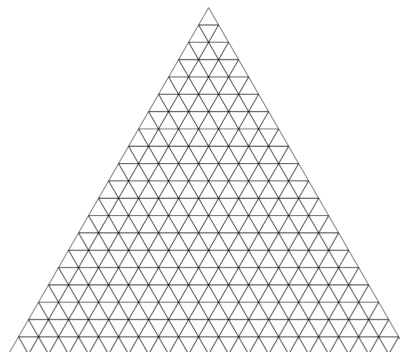


FIG. 2 – *Maillage uniforme raffiné.*

Sur la figure 3, des tailles différentes sont spécifiées aux sommets du triangle.

Enfin, figure 4, des éléments anisotropes sont générés. En chaque sommet (respectivement A, B et C) sont spécifiées une direction (resp. 30°, 150° et 90°), la taille souhaitée le long de cette direction (0.2), et la taille souhaitée perpendiculairement à cette direction (0.02).

Fichier `x.0.0.data`:

```
100   100   100
A -1 0
B  1 0
C  0 1.732
;
;
AB NULL A B NULL ;
BC NULL B C NULL ;
CA NULL C A NULL ;
;
;
;
AB +1 0
;
isotrope
A 0.05   B 0.1   C 0.2 ;
```

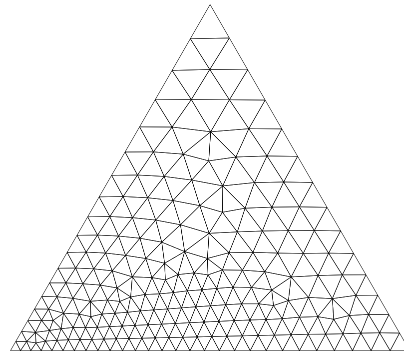


Fig. 3 – *Maillage isotrope.*

Fichier `x.0.0.data`:

```
100   100   100
A -1 0
B  1 0
C  0 1.732
;
;
AB NULL A B NULL ;
BC NULL B C NULL ;
CA NULL C A NULL ;
;
;
;
AB +1 0
;
anisotrope
A  30 0.2 0.02
B 150 0.2 0.02
C  90 0.2 0.02
;
```

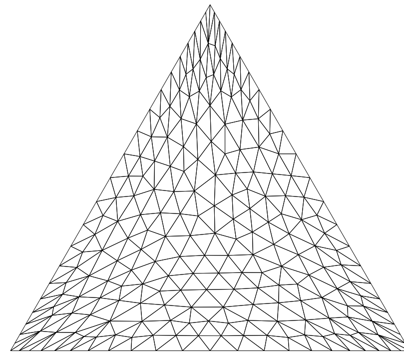


Fig. 4 – *Maillage anisotrope.*

1.2 Maillage adaptatif

Considérons maintenant le cas plus complexe d'un maillage adaptatif. Un exemple typique consiste à définir un domaine du plan et à le mailler, afin de résoudre un problème par la méthode des éléments finis. Le cas échéant, le maillage est raffiné pour améliorer la précision des résultats.

Nous devons tout d'abord définir les données géométriques et physiques du domaine. Sa frontière est définie par des segments droits ou courbes, que nous représentons dans tous les cas par des *splines*. Nous devons ensuite discrétiser les splines, mailler le domaine, et effectuer les calculs par éléments finis. Enfin, selon les résultats obtenus, nous devons adapter le maillage et recommencer les calculs, ou bien arrêter le processus.

En pratique, ceci revient à activer plusieurs programmes exécutables (manuellement ou par un shell-script, comme nous le verrons plus tard). Dans le logiciel BL2D-V2, ces programmes ont pour noms `blg` (définition du domaine de calcul), `blsmo` (calcul des splines), `blmc` (discrétisation des courbes) et `bltms` (maillage des domaines en triangles). Les principales étapes à effectuer sont présentées rapidement ci-dessous (et sont précisées ensuite) :

1. Définir les données géométriques et physiques, ainsi que les données qui gouvernent le premier maillage. Pour cela, deux fichiers initiaux doivent être créés :
   - le fichier `x.0.0.g` (au format *g*), qui contient les données géométriques et physiques d'un domaine du plan,
   - le fichier `x.0.0.hg` (au format *h*), qui permet de gouverner les mailleurs de courbes et de domaines.

   Pour créer ces fichiers, il est possible d'utiliser le programme `blg` qui fait partie du logiciel BL2D-V2. Une autre solution, plus simple pour l'utilisateur, est de disposer d'une interface avec son logiciel habituel de CAO ou de simulation. Les formats de tous ces fichiers sont décrits en section 3.
2. Activer le programme de calcul des "splines" (`blsmo`). Une spline est un segment droit ou courbe défini par des points de contrôle. Le programme `blsmo` sert à approcher ces splines par une discrétisation très fine, qui constitue un support géométrique de la frontière.
3. Activer le discrétiseur de courbes (`blmc`) qui subdivise le support géométrique de la frontière en un ensemble d'arêtes, en respectant la carte de taille spécifiée.
4. Activer le mailleur de domaines (`bltms`) qui subdivise chaque sous-domaine en un ensemble de triangles, et crée ainsi le maillage demandé.
5. Activer le programme de calcul par éléments finis. Si le maillage doit être adapté, créer un nouveau fichier pour gouverner le mailleur et retourner à l'étape de discrétisation des courbes (3). Sinon, le processus est terminé.

Mise en œuvre

Dans cette section, nous présentons en détail la succession d'étapes précédente. Après quelques définitions préalables, nous montrons comment réaliser un maillage initial isotrope, puis des maillages adaptés anisotropes.

Définitions et notations

Nous supposons ici que l'utilisateur travaille sous système Unix (d'autres systèmes comme Windows sont également utilisables). Les fichiers sont organisés de manière hiérarchique en *répertoires*.

Shell

Un *shell* est un interpréteur de commandes. Pour inviter l'utilisateur à taper une commande, il imprime un caractère d'appel (*prompt*) noté ici `%`.

En outre, le *shell* est programmable : un *shell-script* est un programme écrit dans le langage du *shell*.

Répertoire de distribution

Le *répertoire de distribution* contient l'ensemble des programmes et des données du logiciel BL2D-V2. Il est noté `~bl2d` et contient en fait plusieurs sous-répertoires :

- `~bl2d/s` : programmes sources,
- `~bl2d/`*machine* (*machine* = `alpha`, `hp700`, `ibm`, `sun`, ...) : programmes exécutables pour une machine cible particulière (respectivement DEC Alpha, HP série 700, IBM RS6000, Sun 4, ...),
- `~bl2d/data` : données fournies à titre d'exemple.

Il est recommandé d'ajouter le sous-répertoire `~bl2d/`*machine* à la variable d'environnement `path` (qui indique au *shell* où trouver les commandes).

Répertoire de travail

Le *répertoire de travail* est un espace disque quelconque où l'utilisateur peut créer des fichiers. Il est noté ici `/tmp`, mais il est évidemment possible d'utiliser un répertoire personnel créé par la commande `mkdir`.

Création d'un maillage initial isotrope

Ici, les données initiales sont spécifiées par un fichier de texte créé par l'utilisateur (étape 1 ci-après). Pour construire le maillage associé, il suffirait de lancer le shell-script `bl2d.sh`. Cependant, pour des raisons pédagogiques, les différents programmes appelés par ce shell-script sont activés manuellement ci-dessous (étapes 2 et suivantes).

1. Aller dans le répertoire de travail `/tmp`, et y créer le fichier `x.0.0.data` qui est imprimé à la fin de la section 3.2. Ce fichier est normalement distribué avec le logiciel BL2D-V2,

et il suffit alors de le copier (voir les commandes Unix ci-dessous). Sinon, il est assez rapide de le taper à nouveau en sachant qu'il est en format libre [6] : le nombre exact d'espaces entre les données n'est pas important, et tout texte qui commence par les deux caractères `--` est un commentaire ignoré par le logiciel.

```
% cd /tmp
% cp ~bl2d/data/quart/x.0.0.data .
% emacs x.0.0.data &
```

2. Activer le programme `blg` (cf. section 2.4). Le programme lit le fichier `x.0.0.data` et écrit les deux fichiers initiaux `x.0.0.g` et `x.0.0.hg` (Fig. 5) :

   – Le fichier `x.0.0.g` (données géométriques et physiques) décrit un domaine dont la frontière est représentée par 5 splines. Les splines (1) et (5) sont des segments de droites de longueur 1. Les splines (2) et (4) sont des segments de droites de longueur 0.5. La spline (3) est définie par les points de contrôle 5, 6, 7, 8, 9 dans le but d'approcher un quart de cercle. Le point 4 est un point imposé à l'intérieur du domaine (ce point sera obligatoirement un sommet de triangle dans tous les maillages de domaines).
   – Le fichier `x.0.0.hg` (informations pour gouverner le mailleur) demande une taille h = 0.1 aux points 4 (imposé), 1, 2, 6, 8, 3 (extrémités) et 7 (point de contrôle interne à une spline).

3. Facultatif : activer le programme de tracé `bldraw` (cf. section 2.2). Les données sont dirigées par des menus. Le premier menu est le choix du terminal de sortie graphique. En donnant une ligne vide, on obtient un terminal par défaut. En tapant ensuite les données ci-dessous (qui sont expliquées dans la section 2.2), on obtient un tracé à partir du fichier initial `x.0.0.g` (Fig. 5 en haut).

```
% bldraw
...
 Files:  g | hg | smo | mc | hc | ms | ms- | bl2d | hs | h
 Others: b base | z zoom | t test | q quit bldraw
g
 p points (all) | ce constrained+end points | s splines (control polygon) | Return quit draw_g
s
 d draw | n numbers | Return quit draw_gs
d
 d draw | n numbers | Return quit draw_gs

 p points (all) | ce constrained+end points | s splines (control polygon) | Return quit draw_g

 Files:  g | hg | smo | mc | hc | ms | ms- | bl2d | hs | h
 Others: b base | z zoom | t test | q quit bldraw
q
```

4. Activer le programme de calcul des splines `blsmo` (cf. section 2.9).
5. Activer le mailleur de courbes `blmc` (cf. section 2.8).

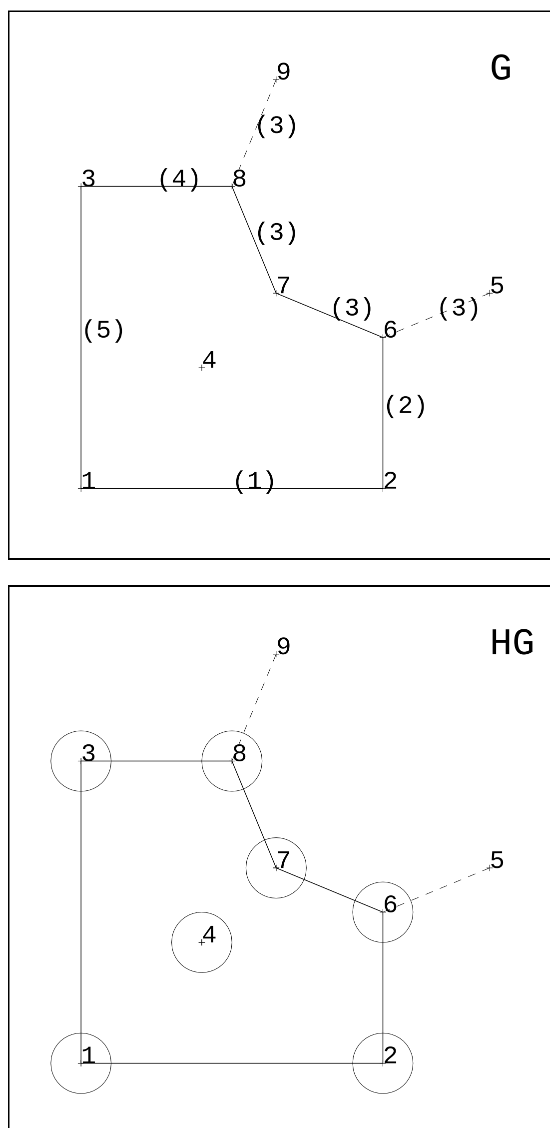


FIG. 5 – *Tracés à partir des fichiers initiaux* `x.0.0.g` *et* `x.0.0.hg`*.*

6. Activer le mailleur de domaines `bltms` (cf. section 2.10).
7. Le maillage initial isotrope du domaine est obtenu. En activant à nouveau le programme `bldraw` (cf. étape 4) avec le dialogue ci-dessous, on peut visualiser ce maillage initial (FIG. 6, $i = 0$).

```
% bldraw
...
 Files:  g | hg | smo | mc | hc | ms | ms- | bl2d | hs | h
 Others: b base | z zoom | t test | q quit bldraw
ms
 phe phys. ref. of edges | phd phys. ref. of subdomains | nd numbers associated with subdomains
 p points | f faces | cf colored faces | Return quit draw_ms
f
 numero de domaine min (0 boite+objet, 1 objet seulement) ?
1
 numero de domaine max (grand nombre => tout l'objet) ?
1
...
```

Au cours des étapes précédentes, les programmes du logiciel BL2D-V2 créent des fichiers dans différents formats (cf. sections 2 et 3). Dans la mesure du possible, les programmes ne dupliquent pas les informations déjà existantes, et ainsi chaque fichier créé ne contient que des informations nouvelles. Les principaux fichiers créés sont le maillage de courbes (format MC) et le maillage de domaines (format MS). On peut par exemple extraire une partie du fichier initial G et des nouveaux fichiers C et MS pour obtenir un fichier au format AMDBA. Ce dernier regroupe un certain nombre de données nécessaires à un programme de calcul par éléments finis (cf. programme `blcv` section 2.1 et format AMDBA section 3.1).

Création de maillages adaptés anisotropes

Après le premier calcul par éléments finis, il est possible d'adapter le maillage et d'effectuer un nouveau calcul, par exemple pour améliorer la précision des résultats. Dans ce cas, les résultats sont habituellement analysés par un programme appelé *estimateur*. Ce dernier crée un fichier, par exemple une carte de tailles ou de métriques, qui gouverne le mailleur dans une nouvelle itération. Ici, nous avons simulé cette étape à l'aide du programme `blh` (cf. section 2.5). Ce programme contient plusieurs cas tests prédéfinis, dont le cas numéro 5 qui est défini de la façon suivante :

```
case(5)
   theta = atan2(y-1, x-1)
   h1 = 0.4*abs((x-1)**2 + (y-1)**2 - (0.75)**2) + 0.003
   h2 = 0.1
```

Les trois variables `theta`, `h1` et `h2` définissent une métrique en tout point (`x`, `y`). L'ensemble des points situés à une distance 1 du point (`x`, `y`), dans cette métrique, est une ellipse d'inclinaison `theta` et de tailles `h1` et `h2` selon les deux directions principales (cf. section 3.4).

Les expressions de `theta`, `h1` et `h2` sont telles que, sur le cercle de centre (1,1) et de rayon 0.75, on obtient des triangles étirés tangents à ce cercle, dans un rapport de $0.003/0.1 = 3\%$.

Pour créer des maillages adaptatifs, il suffit d'appeler plusieurs fois le shell-script `bl2d.sh` jusqu'à ce que maillage soit jugé satisfaisant.

```
% bl2d.sh -exec a -adapt 1
Test number?
5

% bl2d.sh -exec a -adapt 2
Test number?
5

% bl2d.sh -exec a -adapt 3
Test number?
5
```

Bien sûr, un nouveau shell-script peut aussi réaliser cette boucle d'adaptation :

```
#!/bin/sh
for i in 1 2 3
do bl2d.sh -exec a -adapt $i <<!
5
!
done
```

Une dernière possibilité est d'activer manuellement le programme `blh`, de créer un fichier `bl2d.env` contenant la ligne `adapt 1`, et d'activer une nouvelle fois, les programmes `blmc` et `bltms`. Il en résulte un premier maillage adapté, et la boucle peut se poursuivre (avec un numéro d'adaptation égal à 2, 3, ...).

Quelle que soit la possibilité choisie, on obtient trois nouveaux maillages (FIG. 6, $i = 1$ à 3). Les maillages sont de plus en plus conformes à la métrique souhaitée. Remarquons que tous les maillages respectent exactement le point imposé de coordonnées (0.4, 0.4). Remarquons aussi que certaines arêtes ont leurs deux extrémités sur la frontière. Si ceci n'est pas souhaité, ce qui est généralement le cas pour des calculs par éléments finis, il suffit d'ajouter au fichier `bl2d.env` la ligne `rigid 1`.

Gestion des fichiers

Au cours des étapes précédentes, les différents programmes du logiciel BL2D-V2 se transmettent des données par l'intermédiaire de fichiers. La gestion de ces fichiers est pratiquement transparente pour l'utilisateur grâce à une convention simple sur leurs noms, qui sont toujours de la forme :

base.geom.adapt.format

- *base* est un nom de base, ou préfixe, choisi par l'utilisateur. Par défaut, le nom de base est `x`.

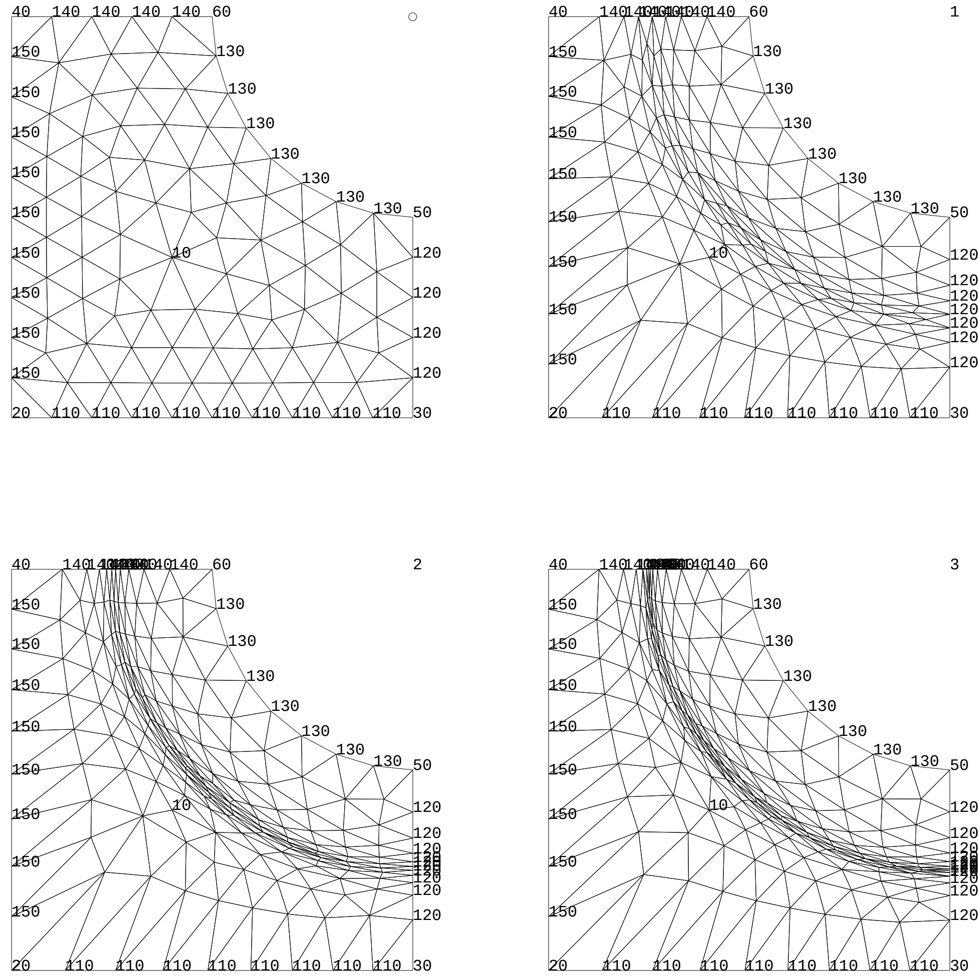


FIG. 6 – *Maillage initial ($i = 0$) et maillages adaptés ($i = 1$, 2 et 3).*

- *geom* est un numéro qui permet de prendre en compte les changements de formes géométriques de la frontière (pour les maillages mobiles).
- *adapt* est le compteur de la boucle d'adaptation, qui vaut 0 lors de la création du maillage initial (non adapté) et est incrémenté de 1 à chaque itération.
- *format* indique le format du fichier (cf. section 3).

1.3 Maillage mobile

...

1.4 Fichier d'entrée `bl2d.env` : variables d'environnement

Au cours des exemples précédents, le fichier `bl2d.env` a été cité à plusieurs reprises. En effet, le logiciel BL2D-V2 est gouverné par un ensemble de variables d'environnement. Chaque variable a un nom prédéfini et une valeur par défaut. Le fichier `bl2d.env` permet de modifier les valeurs de ces variables.

Un exemple simple (généré par `bl2d.sh`) est donné ci-dessous :

```
verb    100
adapt   0
element p1
pref    x
blh     -1
frontal 1
rigid   0
```

Les variables disponibles sont les suivantes :

```
LSS
adapt
angle_corner
angle_geom
angle_smo
base (anciennement pref)
bg
blh
blsmo
box
coiter
delta
div_h
element
eps_smo
file_h
file_interpol
formatted
front_file
frontal
geom
geom_flag
```

```
hmax
hmaxrel
hmin
hminrel
insert
interpol_flag
length_file
maxc
maxs
medial_axis
memory
metric_flag
metric_interpol
periodic
refs
rigid
shockmax
smooth
verb
welldef
```

...

2 Les programmes par ordre alphabétique

La description détaillée des programmes que les utilisateurs peuvent activer est donnée ci-dessous par ordre alphabétique.

2.1 Programme blcv

Le programme `blcv` permet de convertir divers types de fichiers, comme l'indique son menu :

```
 1 .g .mc .ms      -> AMDBA
 2 .g .mc .ms      -> AM
 3 .g .mc .ms      -> AM_FMT
 4 .g .mc .ms      -> .bl2d   ! blexport
 5 MESH            -> OFF
 6 MESH            -> POINTS, FACES
 7 INP             -> MESH
 8 VERT, FACE      -> MESH
 9 MS              -> AMDBA with references set to 0
10 MESH            -> AMDBA
11 MESH (0 based) -> MESH (1 based)
12 WRL             -> MESH
13 G, MC, MS       -> NOPOF (formatted) with triangles
14 G, MC, MS       -> NOPO with triangles
15 G, MC, FMT      -> NOPO with quadrangles
16 AMDBA           -> AM_FMT
17
18 .bl2d           -> .g .hg .smo .mc .ms-   ! blimport
```

```
19
20 AMDBA          -> MESH
21
22 TRI            -> MESH
23 .points .faces -> .mesh
24 .nopo          -> .0.0.bl2d   ! 2d triangles only
25 .bl2d          -> .mesh
26 .mesh          -> .stl
27 .mesh          -> .mesh after symmetry and rotation (e.g. Columbia, 747)
28 .mesh          -> .wrl
29 .mesh          -> .nopo 2D with triangles (ISMC)
30 .mesh          -> number of elements
31 .mesh          -> .points .faces (no allocation)
32 .mesh special  -> .blm (molecule)
```

2.2 Programme `bldraw`

Données :	dirigées par des menus
Fichiers d'entrée :	selon le tracé à effectuer
Fichiers de sortie :	écran ou PostScript

Le programme `bldraw` réalise des tracés à partir de différents fichiers. Il utilise la bibliothèque graphique Fortran 3D, qui permet un choix dynamique du terminal de sortie graphique [7, partie 3]. L'utilisateur est constamment guidé par des menus. Le menu principal est le suivant :

```
Fichiers : g | hg | smo | c (mc) | hc | s (ms) | hs | h
Autres : b base | t test | q quitter bldraw
```

Le menu `Fichiers` permet de choisir un ou plusieurs fichiers à tracer (cf. formats définis en section 3) :

- `g` : données géométriques et physiques initiales (fichier G),
- `hg` : carte H initiale (fichier HG),
- `smo` : splines créées par le programme `blsmo`,
- `c` ou `mc` : maillage de courbes créé par le programme `blmc`,
- `hc` : carte H créée par le programme `blmc`,
- `s` ou `ms` : maillage de domaines créé par le programme `bltms`,
- `hs` : carte H créée par le programme `bltms`,
- `h` : carte H créée par le programme `blh` ou par un *estimateur*.

Le menu `Autres` permet d'accéder à d'autres fonctionnalités :

- `b` : fonctions de base (effacer l'écran, changer la hauteur des caractères ou l'épaisseur des traits, afficher un texte, faire un zoom, ...),
- `t` : tests divers,
- `q` : quitter le programme `bldraw`.

Par exemple, en tapant `g` et en répondant aux menus qui apparaissent successivement, on obtient un tracé à partir d'un fichier G (Fig. 5 en haut).

2.3 Programme `blexport`

...

2.4 Programme `blg`

Fichiers d'entrée :	x.0.0.data	
Fichiers de sortie :	x.0.0.g	x.0.0.hg

Le programme `blg` permet de créer les fichiers initiaux G et HG (cf. section 1.2). Il utilise un fichier de données créé par l'utilisateur (cf. section 3.2). Un autre moyen de créer ces fichiers initiaux est d'utiliser un logiciel interne spécialisé pour certaines applications (par exemple, dans le domaine de la métallurgie, calcoMESH [8]).

2.5 Programme `blh`

Fichiers d'entrée :	x.0.i.c	x.0.i.ms
Fichiers de sortie :	x.0.i.h	

Le programme `blh` simule un *estimateur*. Habituellement, un *estimateur* analyse le résultat d'un calcul par éléments finis et en déduit une carte de tailles ou de métriques. Ici, la carte est définie analytiquement par l'utilisateur, sous forme d'instructions Fortran.

Par exemple (cf. section 1.2), le cas numéro 5 du programme `blh` est reproduit ci-dessous. Sur le cercle de centre (1,1) et de rayon 0.75, il définit des triangles étirés tangents à ce cercle, dans un rapport de $0.003/0.1 = 3\%$.

```
case(5)
   theta = atan2(y-1, x-1)
   h1 = 0.4*abs((x-1)**2 + (y-1)**2 - (0.75)**2) + 0.003
   h2 = 0.1
```

Il est facile d'ajouter de nouveaux cas dans le programme source `blh.f90` présent dans le répertoire de distribution (cf. section 1.2).

2.6 Programme `blimport`

...

2.7 Programme `blinterpol`

Fichiers d'entrée :	x.0.i.bb	x.0.i+1.is
Fichiers de sortie :	x.0.i.bbi	

Le programme `blinterpol` utilise deux fichiers d'entrée qui contiennent respectivement :
- la solution d'un calcul par éléments finis sur un maillage i,
- les coordonnées barycentriques des points du maillage $i+1$ par rapport au maillage i (cf. section 3.5).
  Il en déduit une solution interpolée qui permet d'initialiser le calcul suivant (la version fournie réalise une interpolation de type P1 mais peut être modifiée).

2.8 Programme `blmc`

Fichiers d'entrée :	si $i = 0$: x.0.0.g	x.0.0.smo	x.0.0.hg	
	si $i \geq 1$: x.0.0.g	x.0.0.smo	x.0.i-1.c	x.0.i-1.h
Fichiers de sortie :	si $i = 0$: x.0.0.c	x.0.0.hc		
	si $i \geq 1$: x.0.i.c	x.0.i.hc	x.0.i.ic	

Le programme `blmc` est un mailleur de courbes : il discrétise chaque segment courbe en arêtes. Les tailles de ces arêtes sont déterminées par le fichier d'entrée `x.0.0.hg` si $i = 0$, et `x.0.i-1.h` si $i \geq 1$.

2.9 Programme `blsmo`

Fichiers d'entrée :	x.0.0.g
Fichiers de sortie :	x.0.0.smo

Le programme `blsmo` crée un fichier SMO (cf. section 3.8).

2.10 Programme `bltms`

Fichiers d'entrée :	si $i = 0$: x.0.0.mc	x.0.0.hc		
	si $i \geq 1$: x.0.i.mc	x.0.i.hc	x.0.i-1.ms	x.0.i-1.h
Fichiers de sortie :	si $i = 0$: x.0.0.ms	x.0.0.hs		
	si $i \geq 1$: x.0.i.ms	x.0.i.hs	x.0.i.is	

Le programme `bltms` est un mailleur de domaines : il discrétise chaque sous-domaine en triangles. Les tailles et les formes de ces triangles sont déterminées par le fichier d'entrée x.0.i.hc créé par le mailleur de courbes.

3 Les formats de fichiers par ordre alphabétique

Dans le cadre du mailleur bidimensionnel BL2D-V2, nous avons défini les formats ci-après, classifiés en trois catégories :

1. Les formats d'entrée du logiciel BL2D-V2 :
   - G : données géométriques et physiques,
   - H : carte de tailles ou de métriques.
2. Les formats de sortie du logiciel BL2D-V2 (en fait, dans le cas d'une boucle d'adaptation de maillage, ces formats de sortie deviennent des formats d'entrée à l'itération suivante) :
   - MC : maillage de courbes,
   - MS : maillage de domaines.
3. Les formats auxiliaires. Il s'agit de formats d'interface avec d'autres logiciels (AMDBA et DATA) ou de formats internes au logiciel BL2D-V2 (IS, MC, SMO) :
   - AMDBA : maillage de domaines et physique,
   - DATA : données du préprocesseur `blg`,
   - IS : coordonnées barycentriques, en vue d'une interpolation,
   - MC : maillage de courbes utilisé en entrée du mailleur de domaines,
   - SMO : représentation des splines.

Dans cette organisation en plusieurs formats, la distinction entre les données géométriques et physiques, les maillages de courbes et les maillages de domaines évite la duplication d'informations dans des fichiers séparés. Ceci entraîne des gains en simplicité, en espace disque et en temps d'entrée/sortie.

Remarque commune à tous les formats : les nombres flottants

Tout nombre flottant doit pouvoir être lu dans une variable en double précision dans le langage Fortran 90 ou C. Le nombre de chiffres significatifs dépend de la machine et des programmes utilisés.

– Exemple de lecture en Fortran 90 :

```
double precision d
read (file,*) d
```

– Exemple de lecture en C :

```
double d;
scanf(file, "%lf", d);
```

3.1 Format auxiliaire AMDBA

Objectif

L'objectif du format AMDBA est de décrire un maillage de domaines et ses références physiques. Il est utilisé par plusieurs codes de calcul par éléments finis. Cependant, il est

moins général que les formats G, MC et MS réunis. Notamment, il ne donne pas les références physiques des arêtes et ne permet pas de calculer facilement les coordonnées des nœuds P2.

Ces données sont générées par le programme `blcv`.

Définition du format AMDBA

NP NT
Pour tous les points .... $i = 1..\text{NP}$
$i \quad x_i \quad y_i \quad \varphi_i$
Pour tous les triangles .... $i = 1..\text{NT}$
$i \quad p_{i1} \quad p_{i2} \quad p_{i3} \quad \varphi_i$

Notations

NP	Nombre de points.
NT	Nombre de triangles.

Pour un point $i = 1..\text{NP}$:

x_i y_i	Coordonnées du point.
φ_i	Référence physique du point.

Pour un triangle $i = 1..\text{NT}$:

p_{i1} p_{i2} p_{i3}	Numéros des trois sommets donnés dans le sens direct.
φ_i	Référence physique du triangle.

Exemple de fichier au format AMDBA

Le fichier AMDBA ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
121   202
1   0.4000000   0.4000000   10
2   0.0000000   0.0000000   20
3   1.0000000   0.0000000   30
...
1   69   70   98   200
2   96   92   93   200
3   35   58   57   200
...
```

3.2 Format auxiliaire DATA

Objectif

Rappelons que le programme `blg` permet de créer les fichiers initiaux G (données géométriques et physiques) et HG (carte de tailles ou de métriques) (cf. section 1.2). L'objectif du format DATA est de permettre à l'utilisateur de décrire ces données de façon aussi simple que possible.

Principes généraux

Le format DATA ressemble aux formats G et H mis bout à bout, mais offre les possibilités suivantes :

- Les objets géométriques (points, splines, sous-domaines, ...) peuvent être désignés par des noms (identificateurs) et non par des numéros.
- Les nombres d'objets sont calculés automatiquement.
- Les lectures sont en format libre [6]. Il est donc possible d'y inclure des commentaires et des expressions arithmétiques.
- Les métriques anisotropes sont données sous la forme θ, h_1, h_2 et non pas a, b, c. L'angle θ est donné en degrés. La valeur h_1 (resp. h_2) est la taille souhaitée le long de l'axe d'angle θ (resp. $\theta + 90°$).

Définition du format DATA

NP NS ND
Pour tous les points i :
  id_i x_i y_i
;
Pour tous les points imposés :
  id_i
;
Pour toutes les splines :
  id_i de la spline
  $id.$ du point de contrôle 1 (NULL s'il est absent)
  $id.$ de l'extrémité 1
  $id.$ des sommets intermédiaires
  $id.$ de l'extrémité 2
  $id.$ du point de contrôle 2 (NULL s'il est absent)
  ;
;
Pour tous les points imposés ou extrémités t.q. $\varphi_i \neq 0$:
  id_i φ_i
;
Pour toutes les splines t.q. $\varphi_i \neq 0$:
  id_i φ_i
;
Pour tous les sous-domaines t.q. $\varphi_i \neq 0$:
  id_i d'une spline o_i (+1 ou -1) φ_i
;
`isotrope` ou `anisotrope`
Pour tous les points imposés ou extrémités :

id_i
si `isotrope`: h au point i
si `anisotrope`: θ h_1 h_2 au point i
;

Notations

NP	Nombre <u>maximal</u> de points.
NS	Nombre <u>maximal</u> de splines.
ND	Nombre <u>maximal</u> de sous-domaines.
id_i	Identificateur d'un objet i (point, spline, sous-domaine, ...).
x_i y_i	Coordonnées du point i.
φ_i	Référence physique d'un point, d'une spline ou d'un sous-domaine.

Exemple de fichier au format DATA

Le fichier DATA ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
100 -- maximal number of points
100 -- maximal number of segments
100 -- maximal number of sub-domains

-- points
!X=SQRT(2)/4
A 0 0    B 1 0     C 0 1    P 0.4 0.4    D 1+X 1-X
E 1 0.5  F 1-X 1-X  G 0.5 1  H 1-X 1+X    ;

-- required points
P ;

-- segments
AB NULL A B NULL ;
BE NULL B E NULL ;
EG D E F G H ;
GC NULL G C NULL ;
CA NULL C A NULL ;
;

-- physical references of certain required points or end points
P 10   A 20   B 30   C 40   E 50   G 60   ;

-- physical references of certain segments
AB 110   BE 120   EG 130   GC 140   CA 150   ;

-- sub-domains
AB +1 200 ;

-- map of sizes or metrics
isotropic
A 0.1   B 0.1   C 0.1   P 0.1   E 0.1   F 0.1   G 0.1   ;
```

3.3 Format d'entrée G

Objectif

L'objectif du format G est de décrire les données géométriques et physiques d'un domaine du plan. Ces données sont normalement générées par un système interactif graphique ou par un préprocesseur (cf. section 1).

Définitions préalables

Rappels
Un domaine peut être composé d'un ou de plusieurs sous-domaine(s). Chaque sous-domaine est délimité par sa frontière.

Pour effectuer des calculs par éléments finis, il est nécessaire de mailler le domaine. Plusieurs maillages peuvent être réalisés sur un même domaine (par exemple pour adapter le maillage, ou encore pour résoudre un problème multi-physique). Tout maillage doit respecter les frontières des sous-domaines. On peut également imposer que le maillage s'appuie sur des segments ou des points internes à un sous-domaine (on parle alors de segment imposé ou de point imposé). Un segment interne a au plus une extrémité commune avec un segment de frontière (s'il avait deux extrémités communes, il serait lui-même un segment de frontière).

Par exemple (FIG. 7), le domaine Ω comprend deux sous-domaines Ω_1 et Ω_2. Le sous-domaine Ω_1 a pour frontière la composante connexe $\Gamma_1 \cup \Gamma_2$. Le sous-domaine Ω_2 a pour frontière les deux composantes connexes $\Gamma_2 \cup \Gamma_3$ et Γ_4 (il comporte un “trou”). Il contient le segment imposé S et le point imposé P.

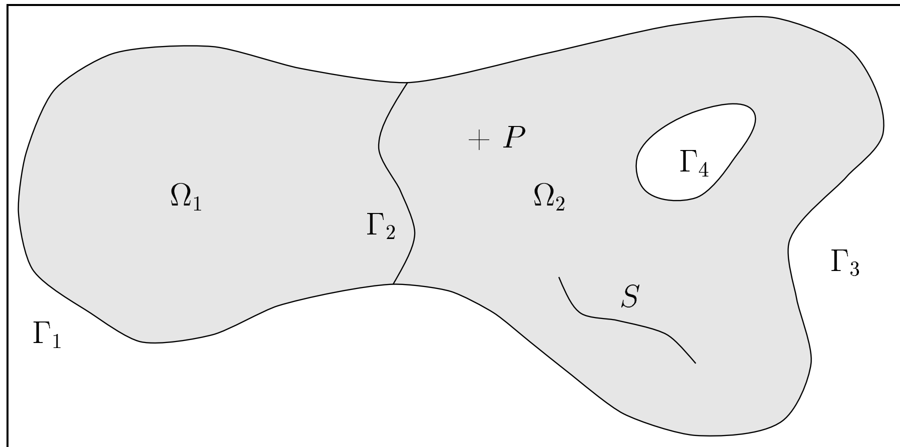


FIG. 7 – *Exemple de domaine* $\Omega = \Omega_1 \cup \Omega_2$.

Des propriétés physiques peuvent être associées aux entités que nous venons de rappeler (sous-domaines, frontières, segments imposés et points imposés). Par exemple, dans le cas de la simulation d'un problème thermique, on peut associer à un sous-domaine sa conductivité et sa source de chaleur, à une frontière son coefficient de transfert et sa température extérieure, et à un point imposé sa température.

Les splines

Il est commode de représenter chaque segment de frontière ou segment imposé par une ou plusieurs spline(s). Une spline seule ne contient pas de point anguleux. En revanche, le point de jonction entre deux splines consécutives peut être anguleux (FIG. 8).

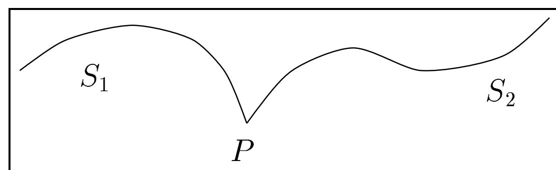


FIG. 8 – *Segment courbe formé de deux splines S_1 et S_2.*

Parmi les nombreuses méthodes utilisées pour définir des splines, nous utilisons celle de Catmull-Rom, où chaque spline est une courbe cubique par morceaux [11].

Par exemple (FIG. 9), la spline $P_2P_3P_4P_5$ est définie par le polygone de contrôle $P_1P_2P_3P_4P_5P_6$. Le morceau de spline P_2P_3 est un morceau cubique défini par ses extrémités P_2 et P_3, par la tangente en P_2 égale à $\overrightarrow{P_1P_3}/2$, et par la tangente en P_3 égale à $\overrightarrow{P_2P_4}/2$. Les morceaux cubiques P_3P_4 et P_4P_5 sont définis de manière analogue.

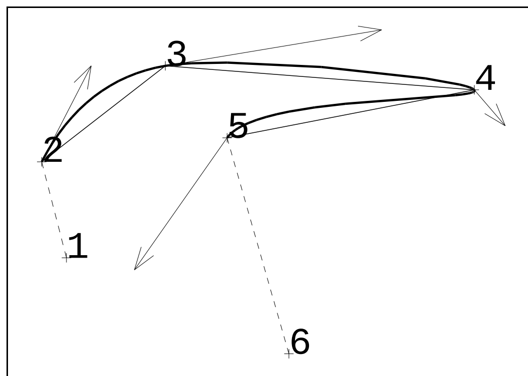


FIG. 9 – *Une spline et son polygone de contrôle.*

De façon plus générale, une spline est décrite par un polygone de contrôle $P_1 \ldots P_n$ ($n \geq 4$, P_1 et P_n facultatifs). Parmi les sommets de ce polygone, on peut distinguer :

- Les extrémités de la spline : P_2 et P_{n-1}.
- Les points de contrôle internes à la spline : $P_3 \ldots P_{n-2}$. Si $n = 4$, ces points sont absents. Sinon ($n > 4$), la spline passe par tous ces points P_i. En chaque point, la tangente aux morceaux $P_{i-1}P_i$ et P_iP_{i+1} est continue et égale au vecteur $\overrightarrow{P_{i-1}P_{i+1}}/2$.
- Les deux points de contrôle externes à la spline : P_1 et P_n. Si le point P_1 (resp. P_n) est présent, il sert à fixer la tangente à l'extrémité P_2 (resp. P_{n-1}) de la spline, en utilisant la méthode précédente. Si un point est absent, la tangente à l'extrémité associée est considérée comme libre.

Ainsi, chaque morceau de spline $P_2P_3, \ldots, P_{n-2}P_{n-1}$ est défini par ses deux extrémités et éventuellement les tangentes en ses deux extrémités. Si les tangentes sont toutes deux fixées, le morceau est cubique (degré ≤ 3). Si une seule tangente est fixée, le morceau est parabolique (degré ≤ 2). Enfin, si aucune tangente n'est fixée, le morceau est linéaire (degré ≤ 1). Il en résulte que, si seules les deux extrémités de la spline sont définies (c'est-à-dire qu'il n'existe aucun point de contrôle, ni interne ni externe), la spline se réduit à un segment de droite.

Les cercles

Le cercle est une figure géométrique fréquemment rencontrée, par exemple dans la représentation de nombreuses pièces mécaniques. Lorsqu'un cercle est approché par une spline, il faut veiller à ce que le nombre n de sommets du polygone de contrôle soit suffisant pour atteindre la précision souhaitée (dans le formalisme précédent, le polygone de contrôle serait noté $P_nP_1P_2P_3 \ldots P_nP_1P_2$, les 3 premiers et les 3 derniers sommets étant confondus). A l'œil nu, le nombre $n = 4$ est insuffisant mais $n = 8$ est déjà acceptable, à condition que les sommets soient équidistants (FIG. 10).

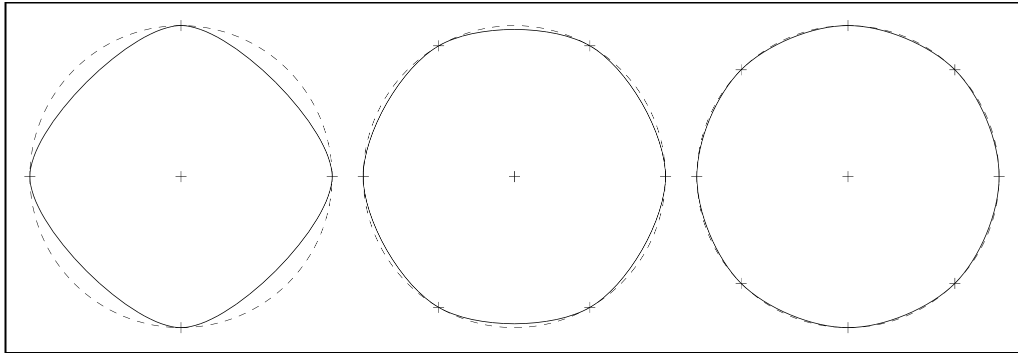


FIG. 10 – *Cercle défini par $n = 4$, 6 ou 8 points.*

Définition du format G

NP NPI NPE NPG NS ND
Pour tous les points .... $i = 1..\text{NP}$
  x_i y_i
Pour tous les points imposés .... $i = 1..\text{NPI}$
  p_i φ_i
Pour toutes les extrémités .... $i = 1..\text{NPE}$
  p_i φ_i
Pour toutes les splines .... $i = 1..\text{NS}$
  np_i
  $(p_{ij})_{\ j=1..np_i}$ φ_i
Pour tous les sous-domaines .... $i = 1..\text{ND}$
  s_i o_i φ_i

Notations

NP	Nombre total de points : NP = NPI + NPE + NPG.
NPI	Nombre de points imposés.
NPE	Nombre d'extrémités.
NPG	Nombre de points restants (ni imposés ni extrémités).
NS	Nombre de splines.
ND	Nombre de sous-domaines.

Pour un point $i = 1..\text{NP}$:

x_i y_i	Coordonnées du point.

Pour un point imposé, $i = 1..\text{NPI}$:

p_i	Numéro du point.
φ_i	Référence physique du point.

Pour une extrémité, $i = 1..\text{NPE}$:

p_i	Numéro du point.
φ_i	Référence physique du point.

Pour une spline $i = 1..\text{NS}$:

np_i	Nombre de points servant à définir la spline ($np_i \geq 4$).
p_{ij}	Numéros des points servant à définir la spline, donnés dans l'ordre suivant :

$p_{i,1}$	Numéro du point de contrôle externe 1. Si ce point est absent, $p_{i,1} = 0$.
$p_{i,2}$	Numéro de l'extrémité 1.
$p_{i,3..np_{i-2}}$	Numéros des points de contrôle internes à la spline, en allant de l'extrémité 1 vers l'extrémité 2.
$p_{i,np_{i-1}}$	Numéro de l'extrémité 2.
p_{i,np_i}	Numéro du point de contrôle externe 2. Si ce point est absent, $p_{i,np_i} = 0$.

φ_i	Référence physique de la spline.

Pour un sous-domaine i = 1..ND :

s_i	Numéro d'une spline appartenant à la frontière.
o_i	Orientation telle que le sous-domaine soit à gauche de la spline s_i. Si o_i = 1, la spline est parcourue de l'extrémité 1 vers l'extrémité 2. Si o_i = -1, la spline est parcourue dans l'autre sens.
φ_i	Référence physique du sous-domaine.

Remarques

La première boucle i = 1..NP définit les coordonnées de tous les points utilisés dans le format G. Ces coordonnées doivent être toutes différentes.

Par définition, un point est dit :

- imposé s'il apparaît dans la boucle i = 1..NPI. Les numéros p_i des points doivent être tous différents.
- extrémité s'il apparaît dans la boucle i = 1..NPE. Les numéros p_i des points doivent aussi être tous différents. En outre, cette boucle doit contenir tous les points qui apparaissent en tant qu'extrémités dans la description des splines (points $p_{i,2}$ et p_{i,np_i-1}).
- point de contrôle externe s'il apparaît en tant que tel dans la description des splines (points $p_{i,1}$ et p_{i,np_i}).
- point de contrôle interne s'il apparaît en tant que tel dans la description des splines (points $p_{i,3..np_i-2}$).
- avec métrique si le fichier HG définit en ce point une taille ou une métrique fixée ($h_i \neq 0$ ou $a_i \neq 0$, cf. section 3.4).

Les différentes combinaisons possibles sont précisées dans le tableau symétrique ci-dessous (avec l'abréviation p.c. = point de contrôle) :

—	imposé	extrémité	p.c. externe	p.c. interne	avec métrique
imposé	—	interdit	possible	possible	obligatoire
extrémité		—	possible	interdit	obligatoire
p.c. externe			—	possible	inutile
p.c. interne				—	possible
avec métrique					—

La combinaison "imposé - extrémité" est interdite afin de numéroter de façon systématique les points des maillages (cf. sections 3.6 et 3.7).

La combinaison "imposé - p.c. externe" est utile dans la définition de certaines splines.

La combinaison "imposé - p.c. interne" permet de d'imposer un point au mailleur de courbes sans diviser artificiellement une spline.

La combinaison "extrémité - p.c. externe" est utile dans le cas des courbes presque fermées.

La combinaison "extrémité - p.c. interne" est interdite car deux splines ne peuvent être adjacentes que si elles ont des extrémités communes.

La combinaison "p.c. externe - p.c. interne" est utile dans le cas des courbes fermées.

La combinaison "p.c. interne - avec métrique" permet de contrôler plus finement le mailleur de courbes.

Dans la description d'un point, d'une spline ou d'un sous-domaine, on appelle référence physique φ_i un nombre entier qui est utilisé comme index vers les propriétés physiques de cette entité. Par convention, $\varphi_i = 0$ indique qu'aucune propriété physique ne doit être associée à l'entité.

Exemple de fichier au format G

Le fichier G ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
9 1 5 3 5 1
   0.0000000000000000   0.0000000000000000
   1.0000000000000000   0.0000000000000000
   0.0000000000000000   1.0000000000000000
   0.4000000000000000   0.4000000000000000
   1.3535533905932737   0.6464466094067263
   1.0000000000000000   0.5000000000000000
   0.6464466094067263   0.6464466094067263
   0.5000000000000000   1.0000000000000000
   0.6464466094067263   1.3535533905932737

4 10

1 20
2 30
3 40
6 50
8 60

4
0 1 2 0 110
4
0 2 6 0 120
5
5 6 7 8 9 130
4
0 8 3 0 140
4
0 3 1 0 150

1 1 200
```

3.4 Format d'entrée H

Objectif

Un mailleur doit créer des éléments de courbes ou de domaines. L'objectif du format H est de gouverner ce mailleur en donnant, en certains points d'un domaine, la taille souhaitée des éléments à créer au voisinage du point (cas isotrope) ou la métrique au voisinage du point (cas anisotrope). Les points du domaine sont eux-mêmes définis dans le format G (données géométriques et physiques), C (maillage de courbes) ou MS (maillage de domaines).

Ces données sont normalement générées par un préprocesseur (pour contrôler le maillage initial) ou par un *estimateur* (pour adapter le maillage).

Définitions préalables

La carte de tailles ou de métriques est définie sur un ensemble de points, qui sont eux-mêmes donnés dans le format G, MC ou MS.

Si le maillage à créer est isotrope, il suffit de donner en chaque point la taille souhaitée des éléments à générer au voisinage de ce point.

Si le maillage à créer est anisotrope, la méthode retenue consiste à fournir en chaque point une *métrique* dans laquelle la taille souhaitée est égale à l'unité [1, 12]. Une métrique est représentée par une matrice symétrique définie positive à trois coefficients $(a,\, b,\, c)$. En prenant pour origine le point où la métrique est donnée, tout point $(x,\, y)$ situé à une distance 1 de l'origine, dans la métrique $(a,\, b,\, c)$, vérifie l'équation :

$$\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1 \qquad \Longleftrightarrow \qquad a\,x^2 + 2\,b\,x\,y + c\,y^2 = 1$$

Il s'agit de l'équation d'une ellipse centrée à l'origine. Par une rotation d'angle θ, de manière à rendre le repère parallèle à l'un des deux axes de l'ellipse (FIG. 11), l'équation se met sous la forme simplifiée :

$$\frac{X^2}{h_1^2} + \frac{Y^2}{h_2^2} = 1$$

h_1 et h_2 représentent les tailles souhaitées selon deux directions orthogonales, dans la métrique usuelle égale à l'identité. Inversement, si l'on connaît θ, h_1 et h_2, il est facile d'obtenir la métrique $(a,\, b,\, c)$ grâce à la relation :

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} = P \begin{pmatrix} \frac{1}{h_1^2} & 0 \\ 0 & \frac{1}{h_2^2} \end{pmatrix} P^{-1}$$

où P est la "matrice de passage" :

$$P = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad \text{et} \qquad P^{-1} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

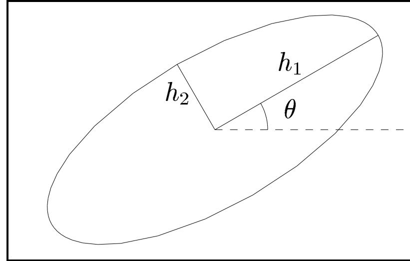


FIG. 11 – *Ellipse définie par θ, h_1 et h_2.*

Définition du format H

NP
`isotrope` ou `anisotrope`
pour chaque point .... $i = 1..\text{NP}$
  si `isotrope` : h_i
  si `anisotrope` : a_i b_i c_i

Notations

NP	Nombre de points où sont définies les tailles ou les métriques.

Pour un point $i = 1..\text{NP}$:

h_i	Taille au point i (cas isotrope).
a_i b_i c_i	Métrique au point i (cas anisotrope).

Remarques

Il est parfois utile de laisser libre la taille ou la métrique en certains points. Par convention, la taille ou la métrique est considérée comme libre en tout point p_i tel que $h_i = 0$ (cas isotrope) ou $a_i = 0$ (cas anisotrope).

Exemple de fichier au format H

Le fichier H ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
9
isotrope
  0.1000000000000000
  0.1000000000000000
  0.1000000000000000
  0.1000000000000000
  0.0000000000000000
  0.1000000000000000
  0.1000000000000000
  0.1000000000000000
  0.0000000000000000
```

3.5 Format auxiliaire IS

Objectif

Au cours d'une boucle d'adaptation, plusieurs maillages d'un même domaine sont générés. L'objectif du format IS est de donner les coordonnées barycentriques des points d'un "nouveau maillage" (*foreground mesh*) par rapport à ceux d'un "ancien maillage" (*background mesh*). Ces données sont créées en vue de l'interpolation de la solution d'un calcul par éléments finis.

Définition du format IS

NP
Pour tous les points du nouveau maillage .... $i = 1..\text{NP}$
p_{i1} p_{i2} p_{i3} λ_{i1} λ_{i2}

Notations

NP Nombre de points du nouveau maillage.

Pour un point du nouveau maillage $i = 1..\text{NP}$:

p_{i1} p_{i2} p_{i3} Numéros de trois points de l'ancien maillage, ou 0 (cf. remarques).

λ_{i1} λ_{i2} Coordonnées barycentriques (cf. remarques).

Remarques

Soit P_i le point numéro i (dans le nouveau maillage).

En général, (p_{i1}, p_{i2}, p_{i3}) sont les numéros des sommets (P_{i1}, P_{i2}, P_{i3}) du triangle de l'ancien maillage qui englobe le point P_i. Alors P_i est défini vectoriellement par :

$$P_i = \lambda_{i1} P_{i1} + \lambda_{i2} P_{i2} + (1 - \lambda_{i1} - \lambda_{i2}) P_{i3}$$

Cependant, il est possible qu'il n'existe pas de triangle englobant (par exemple si une frontière courbe est remaillée, ou si un segment courbe est défini extérieurement au domaine). Dans ce cas, seuls les deux points englobants dans le maillage de courbes sont considérés. Par convention, le numéro du troisième point est nul. Les coordonnées barycentriques sont alors calculées en fonction des abscisses curvilignes. Par exemple, si $p_{i3} = 0$, on aura :

$$\lambda_{i1} = \lambda \qquad \lambda_{i2} = 1 - \lambda \qquad (1 - \lambda_{i1} - \lambda_{i2}) = 0$$

Exemple de fichier au format IS

Le fichier IS ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
204
 49     1    65   0.000000   1.000000
  2    39     0   1.000000   0.000000
  3    16     0   1.000000   0.000000
  4    31     0   1.000000   0.000000
```

```
  5    20   101   1.000000   0.000000
  6   105    26   1.000000   0.000000
  7     8     0   0.672134   0.327866
  8     9    61   0.422772   0.577228
  9    10     0   0.242179   0.757821
...
 30     4    71   0.051587   0.172009
  3    16    48   0.087147   0.073293
  5   101    19   0.457181   0.475098
```

3.6 Format auxiliaire MC

Objectif

L'objectif du format MC est de décrire toutes les données géométriques nécessaires au mailleur de domaines `bltms`. Il décrit donc essentiellement un maillage de courbes. Ces données sont générées par le programme `blmc`.

Définition du format MC

NP NA ND
bb_1 bb_2 bb_3 bb_4
Pour tous les points .... $i = 1..\mathrm{NP}$
x_i y_i
Pour toutes les arêtes .... $i = 1..\mathrm{NA}$
p_{i1} p_{i2}
Pour tous les sous-domaines .... $i = 1..\mathrm{ND}$
q_{i1} q_{i2}

Notations

NP Nombre de points.
NA Nombre d'arêtes.
ND Nombre de sous-domaines.
$bb_1..bb_4$ Bounding box :
(bb_1, bb_2) sont les coordonnées du coin inférieur gauche,
(bb_3, bb_4) sont les coordonnées du coin supérieur droit.

Pour un point $i = 1..\mathrm{NP}$:
x_i y_i Coordonnées du point.

Pour une arête $i = 1..\mathrm{NA}$:
p_{i1} p_{i2} Numéros des deux extrémités de l'arête.

Pour un sous-domaine $i = 1..\mathrm{ND}$:
q_{i1} q_{i2} Numéros des deux extrémités d'une arête frontière ou interne au sous-domaine. L'ordre de ces extrémités est tel que le sous-domaine soit à gauche en allant de q_{i1} vers q_{i2}.

Exemple de fichier au format MC

Le fichier MC ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
39 38 1
0.0000000000000000   0.0000000000000000
1.0000000000000000   1.0000000000000000

0.4000000000000000   0.4000000000000000
0.0000000000000000   0.0000000000000000
1.0000000000000000   0.0000000000000000
...
2 7
7 8
8 9
...
2 7
```

3.7 Format de sortie MS

Objectif

Dans un maillage de domaines, chaque sous-domaine est subdivisé ici en triangles. L'objectif du format MS est de décrire cet ensemble de triangles.

Ces données sont générées par le programme `bltms`.

Définition du format MS

NP NT

Pour tous les points .... $i = 1..\text{NP}$

$x_i \quad y_i$

Pour tous les triangles .... $i = 1..\text{NT}$

$p_{i1} \quad p_{i2} \quad p_{i3} \quad v_{i1} \quad v_{i2} \quad v_{i3} \quad d_i$

Notations

NP Nombre de points.

NT Nombre de triangles.

Pour un point $i = 1..\text{NP}$:

$x_i\ y_i$ Coordonnées du point.

Pour un triangle $i = 1..\text{NT}$:

$p_{i1}\ p_{i2}\ p_{i3}$ Numéros des 3 sommets donnés dans le sens direct.

$v_{i1}\ v_{i2}\ v_{i3}$ Numéros des 3 triangles voisins.

d_i Numéro du sous-domaine :

$d_i = 0$ pour le sous-domaine compris entre l'objet et la *bounding box*,

$1 \leq d_i \leq \text{ND}$ pour les sous-domaines définis dans G,

$d_i \geq \text{ND} + 1$ pour les autres sous-domaines ("trous").

Remarques

Les points sont numérotés de 1 à NP en considérant successivement :

- les NPI points imposés définis dans G,
- les NPE extrémités définies dans G,
- les NPC points créés par le mailleur de courbes (cf. section 3.6),
- les points créés par le mailleur de domaines,
- les 4 points de la *bounding box.*

Les références physiques des points, des arêtes et des triangles, qui sont déjà décrites dans le format G, ne sont pas dupliquées dans le format MS. Elles peuvent être obtenues par des fonctions prévues à cet effet (cf. section 5.2).

Exemple de fichier au format MS

Le fichier MS ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
125 244
 3.99999985098829058e-01   3.99999985098829058e-01
-1.49011709638166786e-08  -1.49011709638166786e-08
 9.99999985098829036e-01  -1.49011709638166786e-08
-1.49011709638166786e-08   9.99999985098829036e-01
 9.99999985098829036e-01   4.99999985098829036e-01
...
 69    70    98    57   128   193   1
 96    92    93   109   235    95   1
125    10     9   160    37    20   0
125    35   124    46     0    42   0
122     5   123    30     0    64   0
...
```

3.8 Format auxiliaire SMO

Objectif

L'objectif du format SMO est de représenter des splines, de manière à calculer rapidement les coordonnées des points appartenant à ces splines.

Définition du format SMO

Pour représenter une spline, il existe essentiellement deux méthodes :

- réaliser un maillage de courbes d'autant plus fin que la courbure de la spline est forte,
- réaliser un maillage de courbes plus grossier, mais y adjoindre certaines informations (coefficients de polynômes cubiques par exemple).

La première méthode a été implémentée dans le logiciel BL2D-V2. En outre, le maillage fin est précédé de sa *bounding box.*

Exemple de fichier au format SMO

Le fichier SMO ci-dessous a été utilisé dans notre exemple. (cf. section 1.2). Le quart de cercle est approché par deux splines cubiques (Fig. 12).

```
   0.0000000000000000    0.0000000000000000
   1.0000000000000000    1.0000000000000000
 37 1 5 31 5
   0.4000000000000000    0.4000000000000000
   0.0000000000000000    0.0000000000000000
   1.0000000000000000    0.0000000000000000
   ...
```

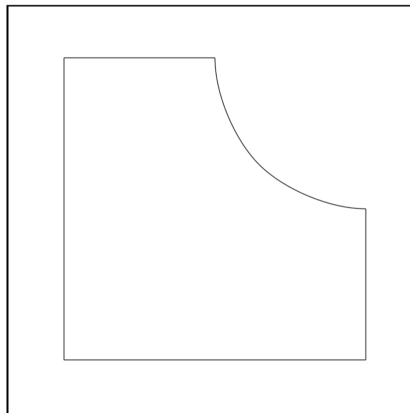


Fig. 12 – *Tracé à partir du fichier* `quart.0.0.smo`.

Partie 2 : manuel de programmation

4 Structures de données

Le logiciel BL2D-V2 est écrit en langages Fortran 90 et C. Il a été installé sur divers types de stations sous système Unix ou Windows. La programmation de ce logiciel est basée sur le concept de *classe*, qui regroupe en une même entité des *structures de données* et des *procédures*. L'implémentation en Fortran 90 est réalisée selon le tableau de correspondance suivant :

Terminologie "objet"	Terminologie Fortran 90
classe	module
– structure	– type dérivé
– procédure	– procédure (subroutine ou fonction)

Cette section décrit les principales structures de données implémentées en Fortran 90. Les procédures, quant à elles, sont décrites dans la section suivante.

Les principaux formats définis précédemment peuvent être représentés en mémoire par des structures de données. Ainsi, aux formats G, MC, MS et H correspondent respectivement les structures g, c, s et h. L'organisation des structures étant très voisine de celle des formats, et les noms des composants des structures étant relativement explicites, seules les déclarations en Fortran 90 sont reproduites ci-dessous.

4.1 Structure g

```
type g_point_
   integer :: ip, physique
end type g_point_

type g_spline_
   integer, dimension(:), pointer :: points
   integer :: physique
end type g_spline_

type g_domaine_
   integer :: is, orientation, physique
end type g_domaine_

type g_
   integer :: np, npi, npe, npg, ns, nd
   double precision, dimension(:,:), pointer :: coor   ! (2,:)
```

```
   type(g_point_),   dimension(:), pointer :: imposes
   type(g_point_),   dimension(:), pointer :: extremites
   type(g_spline_),  dimension(:), pointer :: splines
   type(g_domaine_), dimension(:), pointer :: domaines
end type g_
```

4.2 Structure c

```
type c_spline_
   integer, dimension(:), pointer :: points
   double precision, dimension(:), pointer :: abscisses
end type c_spline_

type c_
   integer :: np, npi, npe, npc, ns
   double precision, dimension(:,:), pointer :: coor   ! (2,:)
   type(c_spline_), dimension(:), pointer :: splines
end type c_
```

4.3 Structure s

```
type s_
   integer :: np, nt
   double precision, dimension(:,:), pointer :: coor   ! (2,:)
   integer, dimension(:,:), pointer :: triangles       ! (3,:)
   integer, dimension(:,:), pointer :: voisins         ! (3,:)
   integer, dimension(:),   pointer :: domaines        ! (:)
end type s_
```

4.4 Structure h

```
type h_
   integer :: np
   character(len=10) :: type_met
   double precision, dimension(:,:), pointer :: mets   ! (3,:)
end type h_
```

5 Procédures

5.1 Lecture et écriture

Les procédures suivantes lisent ou écrivent des structures de types g, c, s ou h définis précédemment :

```
subroutine g_read(g)       subroutine g_write(g)
subroutine c_read(c)       subroutine c_write(c)
subroutine s_read(s)       subroutine s_write(s)
subroutine h_read(h)       subroutine h_write(h)
```

Les extraits qui concernent les allocations de tableaux, à l'intérieur des procédures de lecture, sont reproduits ci-dessous :

```
! structure g_
allocate(g%coor (1:2, g%np))
allocate(g%imposes   (g%npi))
allocate(g%extremites(g%npe))
allocate(g%splines   (g%ns))
allocate(g%domaines  (g%nd))
do i = 1, c%ns
   allocate(g%splines(i)%points(n))
end do

! structure c_
allocate(c%splines(c%ns))
allocate(c%coor(1:2, c%np))
allocate(c%splines(c%ns))
do i = 1, c%ns
   allocate(c%splines(i)%points(n))
   allocate(c%splines(i)%abscisses(n))
end do

! structure s_
allocate(s%coor(1:2, s%np))
allocate(x_s%triangles(3, x_s%nt))
allocate(x_s%voisins  (3, x_s%nt))
allocate(x_s%domaines (   x_s%nt))

! structure h_
allocate(h%mets(3, h%np))
```

5.2 Références physiques

La référence physique d'un point `p`, d'une arête (`p`, `q`) ou d'un triangle `t` est obtenue respectivement par les fonctions `physique_p`, `physique_a` et `physique_d` :

```
function physique_p(p, x_g, x_c) ! p = numero dans le maillage
  integer, intent(in) :: p
  type(g_), intent(in) :: x_g
  type(c_), intent(in) :: x_c
  integer :: physique_p ! intent(out)
end function physique_p

function physique_a(p, q, x_g, x_c)
  integer, intent(in) :: p, q
  type(g_), intent(in) :: x_g
  type(c_), intent(in) :: x_c
  integer :: physique_a ! intent(out)
end function physique_a

function physique_d(t, x_g, x_s)
  integer, intent(in) :: t
  type(g_), intent(in) :: x_g
```

```
    type(g_), intent(in) :: x_s
    integer :: physique_d ! intent(out)
  end function physique_d
```

Les trois fonctions précédentes ont été appelées pour tracer les figures ci-dessous (FIG. 13, 14 et 15).

5.3 Splines

La fonction `eval_s` retourne les coordonnées du point situé sur la spline `is` et d'abscisse curviligne `s` :

```
function eval_s(x_c, is, abscisse)
  type(c_), intent(in) :: x_c
  integer, intent(in) :: is
  double precision, intent(in) :: abscisse
  double precision :: eval_s(2) ! intent(out)
end function eval_s
```

Étant donnée une arête (p, q), la subroutine suivante cherche si cette arête appartient à une spline. Si oui, elle retourne le numéro de la spline et les indices des points `p` et `q`. Sinon, elle retourne 0.

```
subroutine donner_spline(p, q, x_c, is, ip, iq)
  integer, intent(in) :: p, q
  type(c_), intent(in) :: x_c
  integer, intent(out) :: is, ip, iq
end subroutine donner_spline
```

En utilisant les deux procédures précédentes, on peut par exemple calculer les coordonnées du nœud P2 situé au milieu d'une arête (p, q) :

```
call donner_spline(p, q, x_c, is, ip, iq)
if (is == 0) then   ! arete droite
   coor = (x_s%coor(1:2,p) + x_s%coor(1:2,q)) / 2
else                ! arete courbe
   abscisse = (x_c%splines(is)%abscisses(ip) + &
               x_c%splines(is)%abscisses(iq)) / 2
   coor = eval_s(x_smo%c, is, abscisse)
end if
```

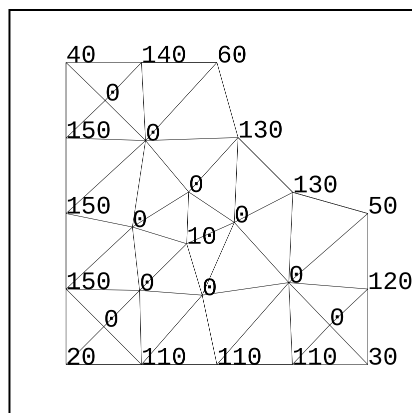


FIG. 13 – *Références physiques des points.*

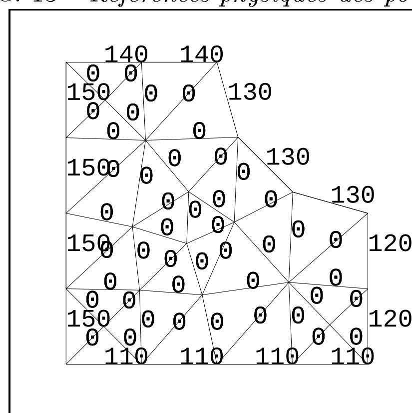


FIG. 14 – *Références physiques des arêtes.*

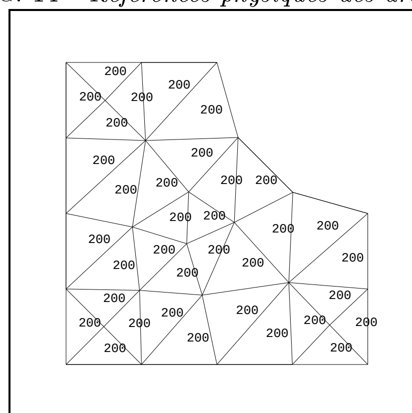


FIG. 15 – *Références physiques des sous-domaines.*

Partie 3 : Exemples d'applications

6 Exemples d'applications

6.1 Arobase

Dans notre premier exemple, la frontière du domaine est un cercle de rayon 50. À l'intérieur, une courbe en forme d'arobase (@) est imposée. La figure 16 représente un maillage "grossier" avec une taille requise de 0.5 sur la courbe imposée, 10 sur le cercle extérieur, et une gradation de 1.4. La figure 17 représente un maillage plus "fin" avec une taille requise de 0.25 sur la courbe imposée, 5 sur le cercle extérieur, et une gradation de 1.25. Sur ces deux figures, la partie gauche (resp. droite) représente le maillage obtenu par une méthode algébrique (resp. frontale). Par ailleurs, la partie inférieure est une vue agrandie de la partie supérieure. La méthode frontale produit un maillage de meilleure qualité (triangles presque équilatéraux, meilleure régularité, degré moins élevé des nœuds, fronts visibles partant de la frontière et de la courbe imposée).

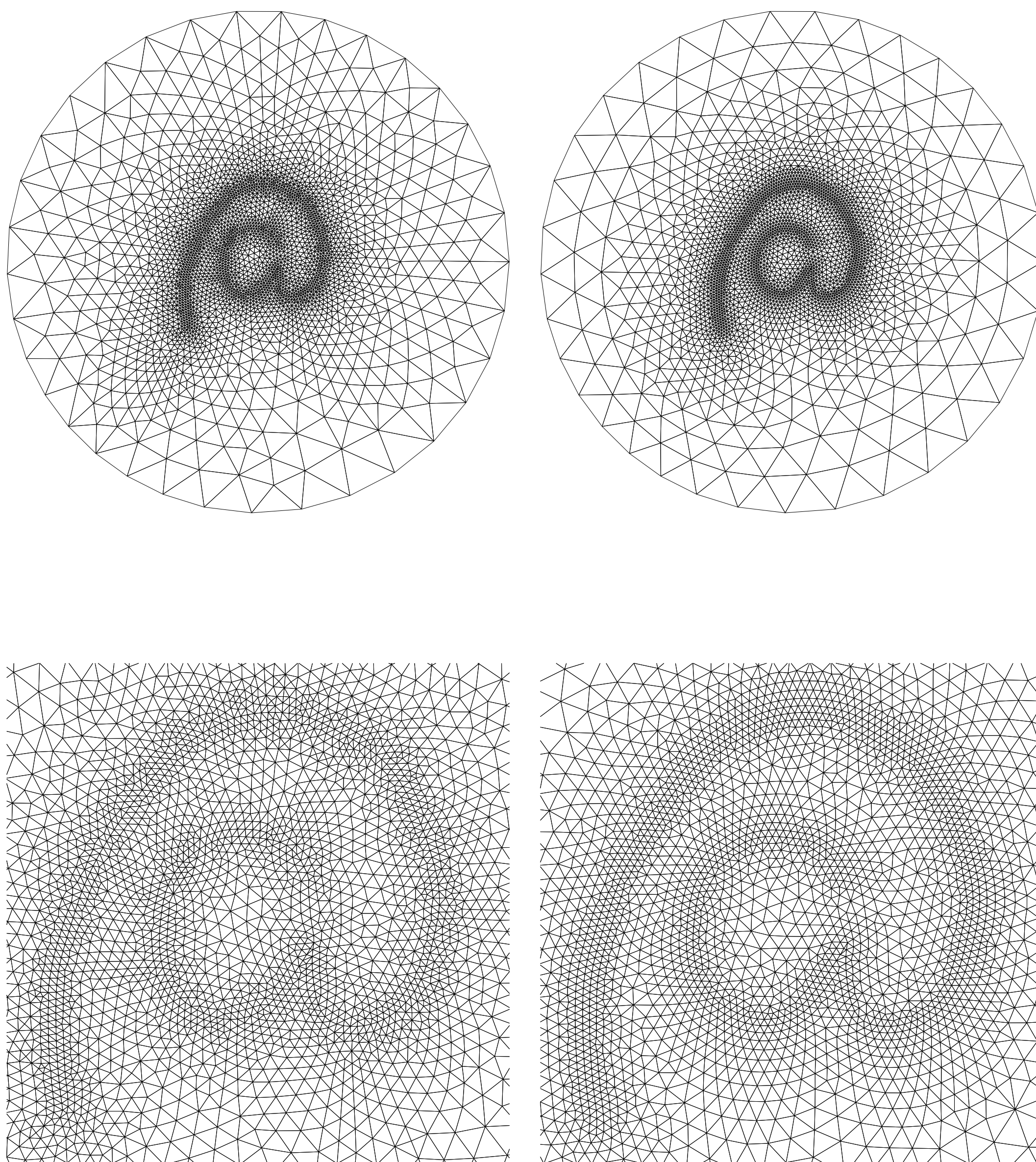


FIG. 16 – *Maillage "grossier". À gauche, méthode algébrique. À droite, méthode frontale. La partie inférieure est une vue agrandie de la partie supérieure.*

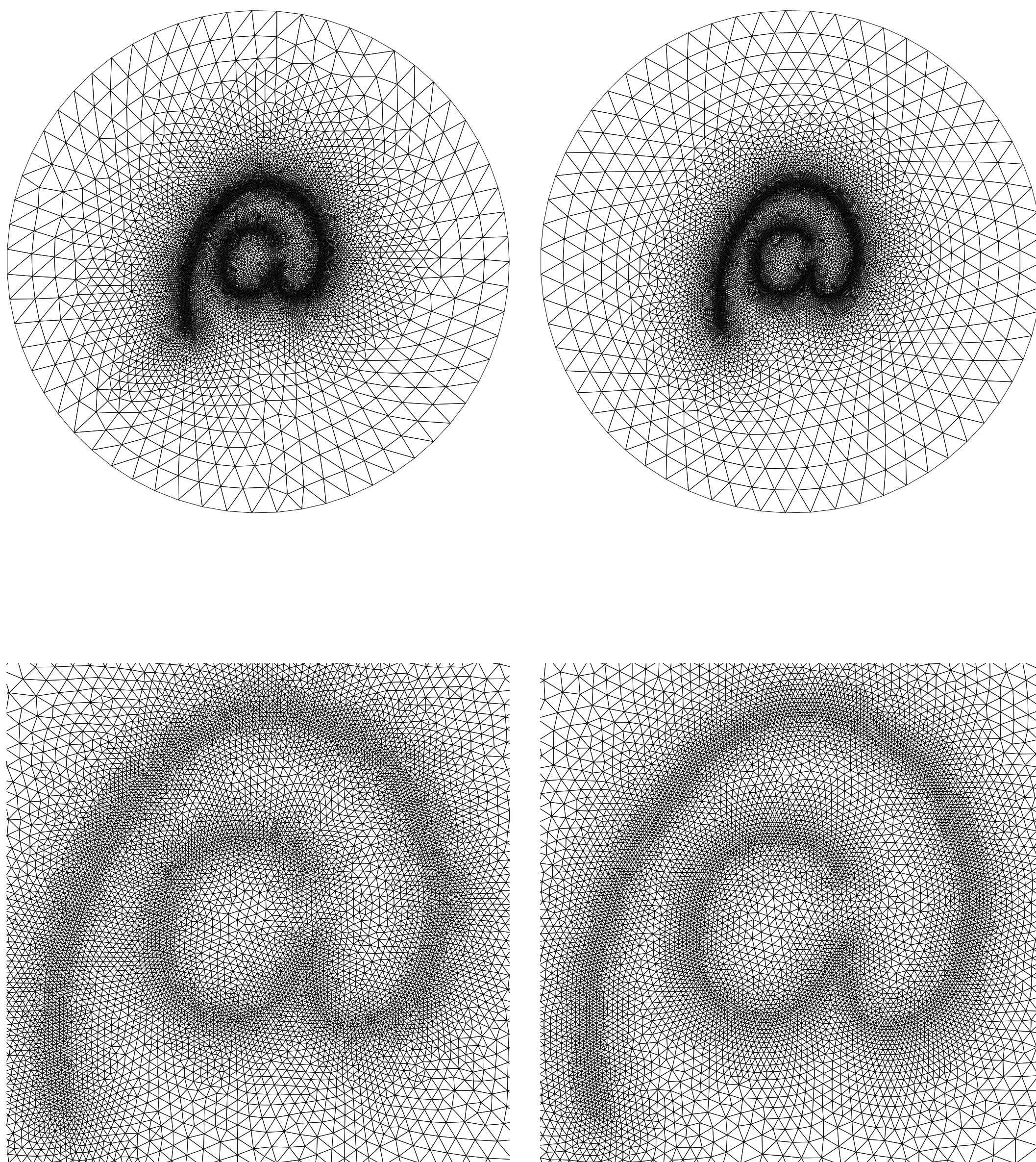


FIG. 17 – *Maillage "fin". À gauche, méthode algébrique. À droite, méthode frontale. La partie inférieure est une vue agrandie de la partie supérieure.*

6.2 Forgeage

Pour illustrer le cas des frontières mobiles, on considère le problème de forgeage d'une pièce mécanique. La pièce et l'outil sont illustrés par la figure 18. L'outil est composé de deux parties, haute et basse. Le procédé consiste à déformer par compression la pièce en descendant la partie haute de l'outil (la partie basse restant fixe) jusqu'à ce que la pièce épouse totalement la forme de l'outil. La déformation de la pièce est obtenue par contact avec l'outil (voir figures 19 à 24).

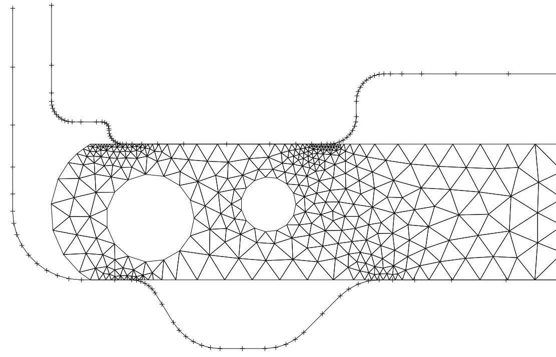


FIG. 18 – *Maillage initial de la pièce*

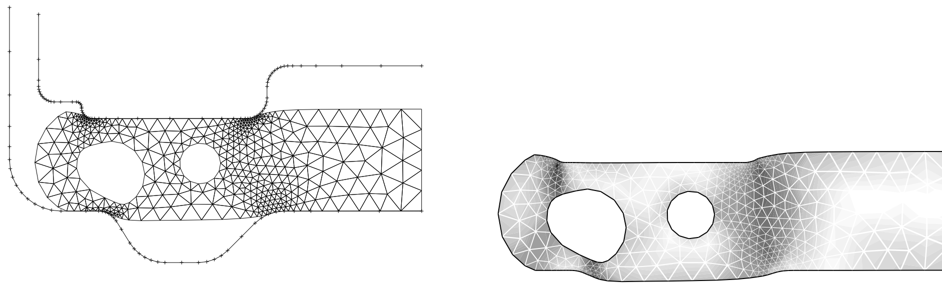


FIG. 19 – *Remaillage et contraintes de Von-Mises à* $\delta = 10$ *mm*

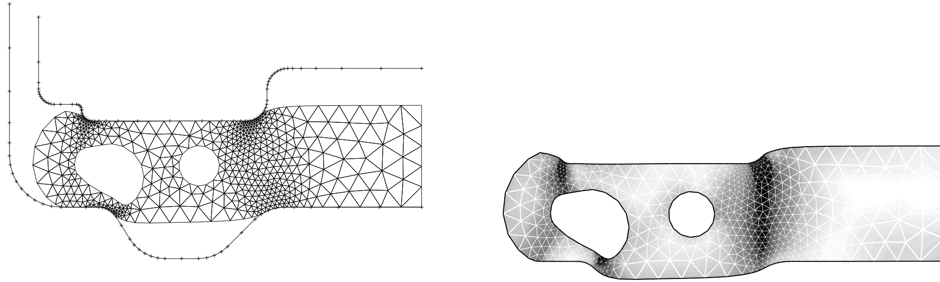


FIG. 20 – *Remaillage et contraintes de Von-Mises à* $\delta = 16$ *mm*

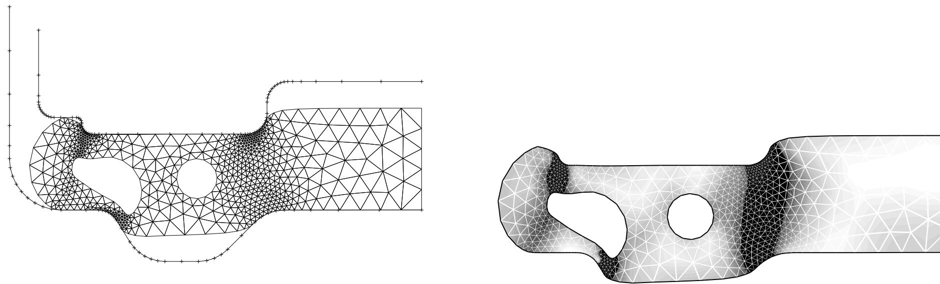


FIG. 21 – *Remaillage et contraintes de Von-Mises à* $\delta = 26$ *mm*

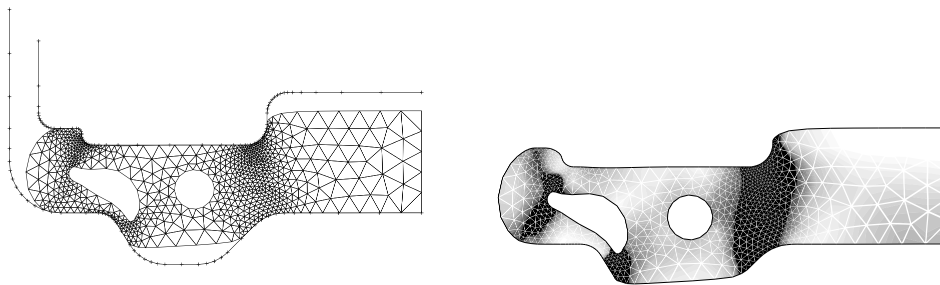


FIG. 22 – *Remaillage et contraintes de Von-Mises à* $\delta = 34$ *mm*

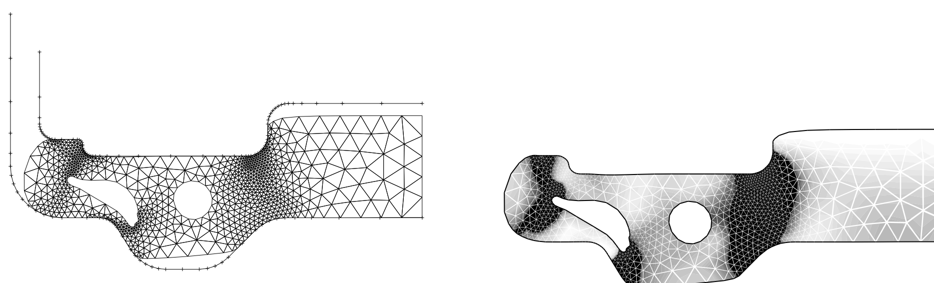


FIG. 23 – *Remaillage et contraintes de Von-Mises à* $\delta = 40$ *mm*

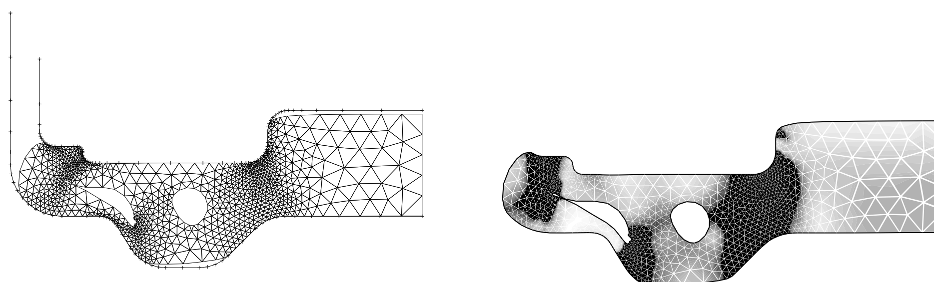


FIG. 24 – *Remaillage et contraintes de Von-Mises à* $\delta = 48$ *mm*

6.3 Quadrilatères

Les figures 25 et 26 montrent des quadrilatères (resp. isotropes et anisotropes) obtenus par appariement.

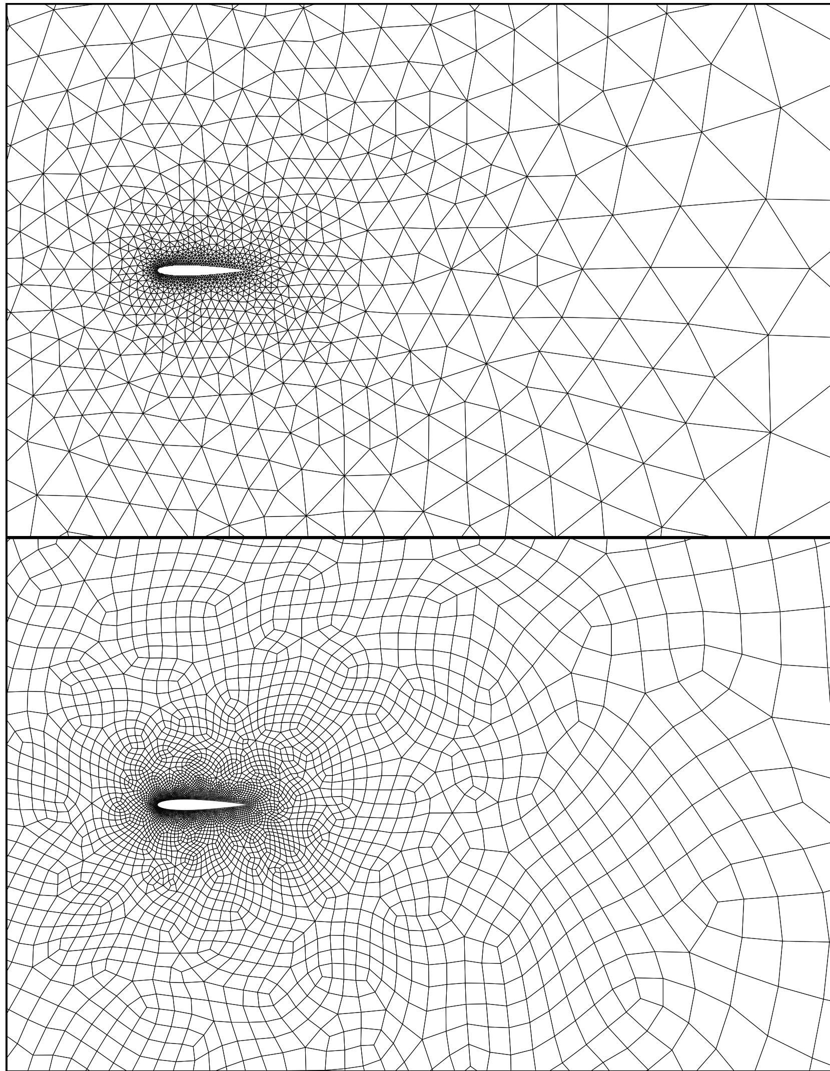


FIG. 25 – *Maillages CFD isotropes : triangles initiaux (en haut) et quadrilatères (en bas).*

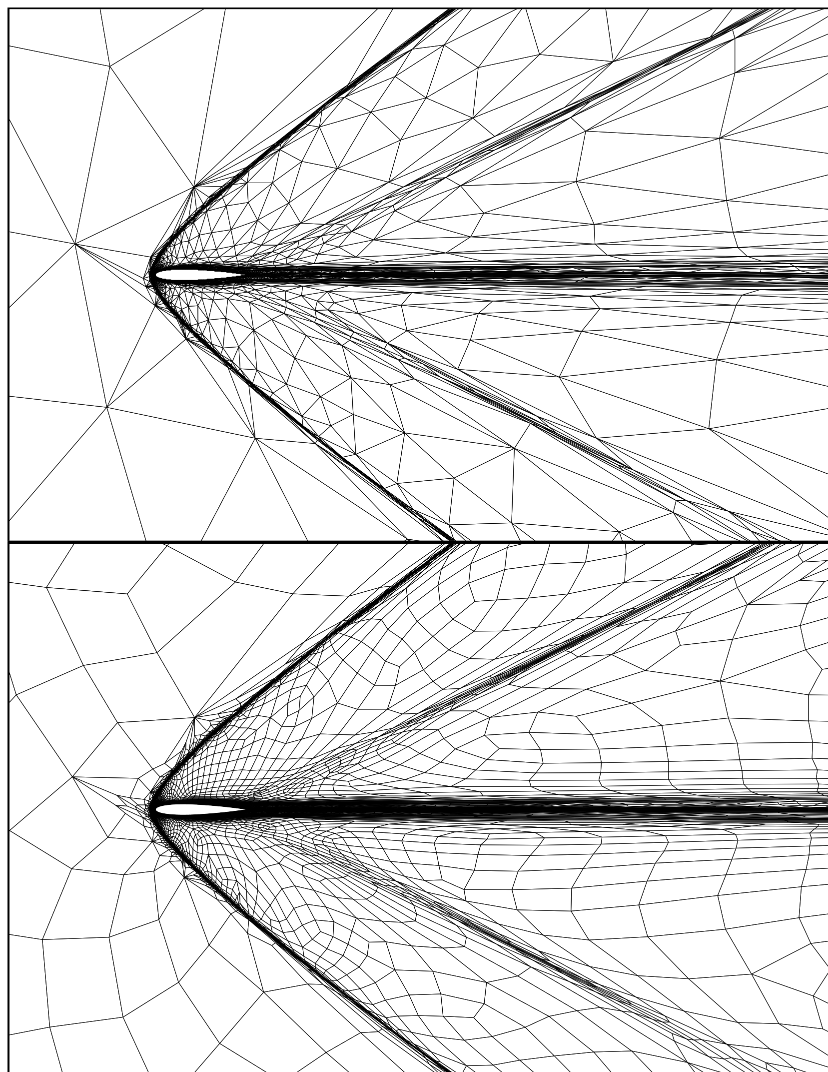


FIG. 26 – *Maillages CFD anisotropes : triangles initiaux (en haut) et quad. (en bas).*

6.4 Photos

Pour conclure sur une note plus distrayante, les figures 27 et 28 montrent des maillages anisotropes adaptés à des photos des auteurs !

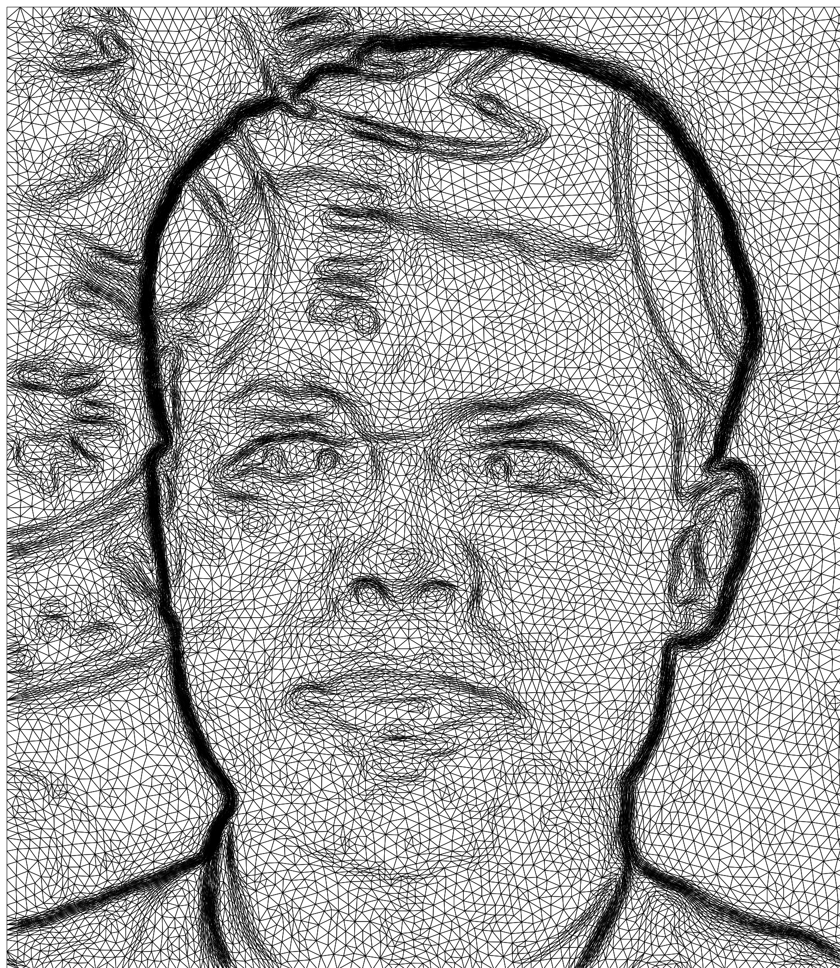


FIG. 27 – *P. Laug : maillage anisotrope.*



FIG. 28 – *H. Borouchaki : maillage anisotrope.*

Références

[1] P. Laug, H. Borouchaki, P.L. George. *Maillage de courbes gouverné par une carte de métriques.* Rapport INRIA RR-2818, mars 1996.

[2] H. Borouchaki, P.L. George, F. Hecht, P. Laug, E. Saltel. *Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I : Algorithmes.* Rapport INRIA RR-2741, décembre 1995.

[3] H. Borouchaki, P.L. George, F. Hecht, P. Laug, B. Mohammadi, E. Saltel. *Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie II : Applications.* Rapport INRIA RR-2760, décembre 1995.

[4] H. Borouchaki, P. Laug. *Le mailleur adaptatif bidimensionnel BL2D : manuel d'utilisation et documentation.* Rapport INRIA RT-0185, décembre 1995.

[5] H. Borouchaki, P. Laug. *The BL2D Mesh Generator: Beginner's Guide, User's and Programmer's Manual.* Rapport INRIA RT-0194, juillet 1996.

[6] P.L. George et P. Laug, *Normes d'utilisation et de programmation*, Guide Modulef n° 2, INRIA, 1992.

[7] P.L. George et E. Saltel, *Post-traitements et graphiques*, Guide Modulef n° 6, INRIA, 1992.

[8] http://www.calcom.ch. CalcoMESH. *Calcom SA, Lausanne, Switzerland.*

[9] http://www-rocq.inria.fr/Patrick.Laug. *Le logiciel BL2D-V2.*

[10] P. Laug, *Rapport intermédiaire "GÉNIE" - Tâches T-2.4.3 à T-2.4.6*, DASSAULT AVIATION - INRIA, 1995.

[11] E. Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, Univ. Utah Comp. Sci. Dept. UTEC-CSC-74-133, 1974.

[12] M.G. Vallet, *Génération de maillages éléments finis anisotropes et adaptatifs*, Thèse Université Paris 6, 1992.



Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803